Claims

1. (Original) A method for profiling an application having plural units, wherein the plural units communicate across strongly-typed, binary-standard interfaces, and wherein a type file describes the interfaces of the plural units, the method comprising:

generating a structural metadata description of the application from the type file, wherein the structural metadata description comprises compiled interface-level type description;

profiling the application using the structural metadata description, resulting in an application profile; and

reconfiguring the application based on the application profile.

2. (Original) The method of claim 1 wherein the step of generating a structural metadata description comprises:

receiving a source code type description of the interfaces;

statically analyzing the source code description; and

producing the structural metadata description of the application.

3. (Original) The method of claim 1 wherein the step of generating a structural metadata description comprises:

receiving a compiled type file comprising information descriptive of the interfaces; and

producing the structural metadata description of the application.

4. (Original) The method of claim 1 wherein the step of profiling comprises:

determining static relationships between the plural units of the application; and

generating the application profile, wherein the application profile models the static relationships.

5. (Original) The method of claim 1 wherein the step of profiling comprises:

determining dynamic interaction between the plural units of the application through interfaces described in the interface-level type description; and

generating the application profile, wherein the application profile models the dynamic interaction.

6. (Original) The method of claim 1 wherein the step of profiling comprises:

measuring the number and size of communications through the interfaces of the plural units using the structural metadata description of the application; and

generating the application profile, wherein the application profile is a log of the communications between the plural units.

7. (Original) The method of claim 1 wherein the step of profiling comprises:

measuring the size of communications through the interfaces of the plural units using the structural metadata description of the application; and

generating the application profile, wherein the application profile is a log of the communications between the plural units.

8. (Original) The method of claim 7 wherein for a communication, the log stores data representing a sending unit, a receiving unit, and the size of the communication.

9. (Original) The method of claim 1 wherein the plural units of the application reside on plural computers in a distributed computing environment, and wherein the step of profiling comprises:

timing communications sent between the plural units; and

generating the application profile, wherein the application profile is a log of the communications sent between the plural units.

10. (Original) The method of claim 1 wherein the step of profiling comprises:

timing the execution of the plural units; and

generating the application profile, wherein the application profile describes the behavior of the plural units.

11. (Original) The method of claim 1 wherein the application is available for profiling only as an application binary.

12. (Original) The method of claim 11 wherein the application binary comprises an executable file.

13. (Original) The method of claim 12 wherein the application binary further comprises one or more dynamic link libraries.

14. (Original) A computer-readable medium having computer-executable instructions for performing the method of claim 1.

15. (Original) The method of claim 1 wherein the step of reconfiguring comprises:

analyzing the application profile; and

modifying the application based on the analysis of the application profile.

16. (Original) The method of claim 1 wherein the step of reconfiguring comprises:

combining the application profile with a network profile;

analyzing the combination of the application and network profiles;

generating a distribution plan; and

during execution of the application, distributing the plural units of the application in a distributed computing environment according to the distribution plan.

17. (Original) The method of claim 16 wherein the network profile comprises a description of an idealized network of plural identical computers with no communication costs.

18. (Original) The method of claim 16 wherein the network profile comprises a description of the capabilities of plural computers in a physical network.

19. (Original) The method of claim 1 wherein the step of reconfiguring comprises:

analyzing the application profile;

generating a distribution plan; and

during execution of the application, distributing the plural units of the application in a distributed computing environment according to the distribution plan.

20. (Original) A method for profiling an application for partitioning and distributing plural units of the application in a distributed computing environment, wherein the plural units

communicate across strongly-typed, binary-standard interfaces, wherein a type file describes the interfaces of the plural units, and wherein the application is available for profiling only as an application binary, the method comprising:

generating a static interface metadata description of the application from the type file, wherein the static interface metadata description comprises type information about the interfaces of the plural units of the application;

profiling the application by using the static interface metadata description on the application binary, resulting in an application profile;

combining the application profile with a network profile;

analyzing the combination of the application and network profiles;

generating a distribution plan; and

during execution of the application, distributing the plural units in the distributed computing environment according to the distribution plan.

21. (Original) The method of claim 20 wherein the step of generating a static interface metadata description comprises:

receiving a source code type description of the interfaces;

analyzing the source code description with static analysis; and

producing the static interface metadata description of the interfaces.

22. (Original) The method of claim 20 wherein the step of generating a static interface metadata description comprises:

receiving a compiled type file comprising information descriptive of the interfaces; and

producing the static interface metadata description of the interfaces.

23. (Original) The method of claim 20 wherein the step of profiling comprises:

determining dynamic interaction between the plural units of the application through interfaces described in the static interface metadata description; and

generating the application profile, wherein the application profile models the dynamic interaction.

24. (Original) The method of claim 20 wherein the step of profiling comprises:

measuring the number and size of communications through the interfaces of the plural units using the static interface metadata description of the application; and

generating the application profile, wherein the application profile is a log of the communications between the plural units.

25. (Original) The method of claim 20 wherein the step of profiling comprises:

measuring the size of communications through the interfaces of the plural units using the static interface metadata description of the application; and

generating the application profile, wherein the application profile is a log of the communications between the plural units.

26. (Original) The method of claim 25 wherein for a communication, the log stores data representing a sending unit, a receiving unit, and the size of the communication.

27. (Original) A computer-readable medium having computer-executable instructions for performing the method of claim 20.

28. (Original) A method for partitioning and distributing plural units of an application in a distributed computing environment, the method comprising:

reading a first set of descriptors describing the application;

reading a second set of descriptors including measurements of the distributed computing environment;

analyzing the first and second sets of descriptors;

generating a distribution plan for the application, the distribution plan comprising information specifying a partitioning of the plural units for distribution in the distributed computing environment;

executing the application; and

during execution of the application, distributing the plural units in the distributed computing environment according to the distribution plan.

29. (Original) The method of claim 28 wherein the first set of descriptors comprises metadata describing the structure of the application.

30. (Original) The method of claim 28 wherein the first set of descriptors comprises information describing one or more location constraints on the placement of the plural units of the application in the distributed computing environment.

31. (Original) The method of claim 28 wherein the first set of descriptors comprises metadata describing the behavior of the application.

32. (Original) The method of claim 28 wherein the plural units have strongly-typed, binary-standard interfaces, and wherein the first set of descriptors describes communications through these interfaces in a set of one or more profiling scenarios.

33. (Original) The method of claim 28 wherein the second set of descriptors comprises measurements of current capabilities of plural computers of a physical network of computers.

34. (Original) The method of claim 28 wherein the second set of descriptors comprises estimates of average latency and average bandwidth for a physical network.

35. (Original) The method of claim 28 wherein the step of analyzing comprises:

grouping the plural units based on the first and second sets of descriptors according to a grouping scheme.

36. (Original) The method of claim 28 wherein the step of analyzing comprises:

representing the first and second sets of descriptors as a commodity flow model; and

finding a minimum cut/ maximum flow for the commodity flow model.

37. (Original) The method of claim 28 wherein the step of generating a distribution plan comprises:

associating the plural units with one or more locations in the distributed computing environment; and

producing the distribution plan, wherein the distribution plan comprises a mapping of the plural units of the application to locations in the distributed computing environment.

38. (Original) A computer-readable medium having computer-executable instructions for performing the method of claim 28.

39. (Original) The method of claim 28 wherein the second set of descriptors describes a physical network of computers; wherein the step of analyzing comprises finding a minimum cut/ maximum flow for a commodity flow model based on the first and second sets of descriptors, and wherein the distribution plan comprises a mapping of the plural units of the application to one or more computers in the physical network.

40. (Original) A computer-readable medium having computer-executable instructions for performing the method of claim 39.

41. (Original) The method of claim 28 wherein system services of the distributed computing environment support distributing the plural units.

42. (Original) The method of claim 28 wherein a combination of system services from a dedicated system and the distributed computing environment supports distributing the plural units.

43. (Original) The method of claim 28 further comprising:

defining a threshold for execution of the application in the distributed computing environment, the threshold describing an accepted deviation from the expected performance of the application;

during execution of the application, if the threshold is exceeded:

noting the recent behavior of the application in the distributed computing environment;

generating a second distribution plan for the application, the second distribution plan assimilating the recent behavior;

executing the application; and

during execution of the application, distributing the plural units in the distributed computing environment according to the second distribution plan.

44. (Original) The method of claim 43 wherein the step of noting the recent behavior comprises:

generating a new first set of descriptors; and

re-analyzing the first and second sets of descriptors.

45. (Original) The method of claim 43 wherein the step of noting the recent behavior comprises:

generating a new second set of descriptors; and

re-analyzing the first and second sets of descriptors.

46. (New) In a computer system, a computer-implemented method comprising:

reading an application profile describing an application that includes plural software components;

reading a network profile describing a distributed computing environment;

analyzing a combination of the application and network profiles;

generating a distribution plan for the application based at least in part on the combination, the distribution plan comprising information specifying a partitioning of the plural software components for distribution in the distributed computing environment; and

automatically distributing the plural software components in the distributed computing environment according to the distribution plan.

47. (New) The method of claim 46 wherein the application profile is generated by profiling an application binary for the application, and wherein the application profile comprises information describing behavior of the application in one or more profiling scenarios.

48. (New) The method of claim 46 wherein the plural software components are dynamic instances created during execution of the application.

49. (New) The method of claim 46 wherein the plural software components have strongly-typed, binary-standard interfaces.

50. (New) The method of claim 46 wherein the network profile comprises measurements of current capabilities of plural computers.

51. (New) The method of claim 46 wherein the network profile comprises latency and bandwidth estimates.

52. (New) The method of claim 46 wherein system services support automatically distributing the plural software components.

53. (New) The method of claim 46 further comprising:

during execution of the application, switching to a second distribution plan; and

automatically re-distributing at least some of the plural software components in the distributed computing environment according to second distribution plan.

54. (New) A computer-readable medium having stored therein computer-executable instructions for causing the computer system to perform the method of claim 46.

55. (New) In a computer system, a computer-implemented method for partitioning and distributing plural software components of an application without access to or reference to source code for the application, wherein existing executable code for the application comprises plural computer-executable instructions, the method comprising:

reading a structural metadata description for the application;

using the structural metadata description and the existing executable code for the application, profiling the application in one or more scenarios; and

using a distribution plan and the existing executable code for the application, automatically distributing the plural software components in the distributed computing environment according to the distribution plan.

56. (New) The method of claim 55 wherein the profiling comprises:

instrumenting the existing executable code for the application; and

recording information about execution in the one or more scenarios.

57. (New) The method of claim 55 wherein the distributing comprises:

instrumenting the existing executable code for the application; and

instantiating each of the plural software components at one of plural locations in the distributed computing environment according to the distribution plan, wherein the plural software components are dynamic instances created during execution of the application.

58. (New) The method of claim 55 wherein the existing executable code for the application comprises application binary code in an executable file and one or more libraries for execution on a hardware processor.

59. (New) The method of claim 55 wherein the plural software components have strongly-typed, binary-standard interfaces, and wherein the structural metadata description comprises compiled interface-level type description.

60. (New) A computer-readable medium having stored therein computer-executable instructions for causing the computer system to perform the method of claim 55.